CLAIMS

1. A circuit enabling the transfer of data, said circuit comprising:

an input multiplexer receiving a first plurality of input data bytes and a second plurality of input data bytes;

a switching controller coupled to said input multiplexer and controlling the output of said data bytes from said input multiplexer;

delay register circuitry coupled to said input multiplexer and receiving predetermined bytes of said first plurality of input data bytes; and

an output multiplexer coupled to said input multiplexer and said delay register circuitry, said output multiplexer receiving said predetermined bytes of said first plurality of input data bytes and predetermined bytes of said second plurality of input data bytes.

2. The circuit of claim 1 wherein said switching controller receives input alignment information associated with said first plurality of input data bytes.

3. The circuit of claim 2 wherein said input alignment information is based upon data valid information received with said first plurality of input data bytes.

4. The circuit of claim 1 wherein said switching controller further receives destination alignment information.

5. The circuit of claim 1 wherein said output multiplexer further generates output data valid information.

6. A programmable logic device enabling the transfer of data, said programmable logic device comprising:

a pass multiplexer receiving a first plurality of input data bytes and a second plurality of input data bytes;

a delay multiplexer receiving predetermined input data bytes of said first plurality of input data bytes;

a switching controller coupled to said pass multiplexer and said delay multiplexer, said switching controller controlling the output of said data bytes from said pass multiplexer and said delay multiplexer;

a delay register coupled to said delay multiplexer and receiving predetermined bytes of said first plurality of input data bytes; and

an output multiplexer coupled to said pass multiplexer and said delay register, said output multiplexer receiving said predetermined bytes of said first plurality of input data bytes from said delay register and predetermined bytes of said second plurality of input data bytes from said pass multiplexer.

7. The programmable logic device of claim 6 wherein said pass multiplexer comprises X-1 multiplexers where X equals the number of data bytes transferred by said programmable logic device.

8. The programmable logic device of claim 6 wherein said delay multiplexer comprises X-2 multiplexers where X equals the number of data bytes transferred by said programmable logic device.

9. The programmable logic device of claim 6 wherein said output multiplexer comprises X-1 multiplexers where X equals the number of data bytes transferred by said programmable logic device.

10. The programmable logic device of claim 6 wherein said switching controller further receives destination alignment information.

11. A programmable logic device enabling the transfer of data, said programmable logic device comprising:

a pass multiplexer receiving a first plurality of input data bytes and a second plurality of input data bytes;

a delay multiplexer receiving predetermined input data bytes of said first plurality of input data bytes;

a switching controller coupled to said pass multiplexer and said delay multiplexer, said switching controller controlling the output of said data bytes from said pass multiplexer and said delay multiplexer;

a delay register coupled to said delay multiplexer and receiving predetermined bytes of said first plurality of input data bytes;

an output multiplexer coupled to said pass multiplexer and said delay register, said output multiplexer receiving said predetermined bytes of said first plurality of input data bytes from said delay register and predetermined bytes of said second plurality of input data bytes from said pass multiplexer; and

a feedback register coupled to said output multiplexer and providing last valid data information to said switching controller.

12. The programmable logic device of claim 11 further comprising an interface circuit coupled to said output multiplexer, said interface circuit temporarily storing valid data bytes from said output multiplexer.

13. The programmable logic device of claim 12 further comprising a data transfer controller coupled to said interface circuit, said data transfer controller determining when said interface circuit is fully loaded.

14. The programmable logic device of claim 13 wherein said data transfer controller generates a register full signal when said interface circuit is fully loaded.

15. The programmable logic device of claim 11 further comprising a transfer signal coupled to said data transfer

controller, said transfer signal automatically transferring the data in said interface circuit.

16. A method of realigning data, said method comprising the steps of:

determining a data alignment of input data comprising a plurality of input bytes;

configuring hardware to selectively transfer input data;

realigning said input data in the hardware based upon said data alignment of said input data; and

outputting said realigned data.

17. The method of claim 16 wherein said step of determining a data alignment of input data comprises a step of determining misaligned data.

18. The method of claim 16 wherein said step of configuring hardware to selectively transfer input data comprises a step of configuring programmable hardware to generate an arbitrary byte alignment of said output.

19. The method of claim 16 wherein said step of configuring hardware to selectively transfer input data comprises a step of configuring programmable hardware to generate a fixed byte alignment of said output.

20. The method of claim 16 further comprising a step of concatenating a second plurality of input bytes with said plurality of input bytes.

21. A method of realigning data, said method comprising the steps of:

receiving a plurality of data bytes having an input alignment;

receiving destination alignment information;

generating a plurality of control signals based upon said input alignment and said destination alignment information;

controlling a plurality of multiplexers based upon said control signals to realign said received data; and

outputting said realigned data.

22. The method of claim 21 wherein said step of controlling a plurality of multiplexers based upon said control signals comprises coupling predetermined bytes of said plurality of data bytes to a delay register.

23. The method of claim 22 further comprising a step of coupling a second plurality of bytes to an output register.

24. The method of claim 22 further comprising a step of transferring said plurality of data bytes in said delay register to said output register.

25. The method of claim 21 further comprising a step of generating output data valid information.

26. A method of realigning data, said method comprising the steps of:

receiving a plurality of data bytes;

receiving input alignment information based upon the alignment of said plurality of data bytes;

receiving destination alignment information;

generating a plurality of control signals based upon said input alignment information and said destination alignment information;

coupling predetermined bytes of said first plurality of bytes to a delay register;

coupling predetermined bytes of a second plurality of data bytes to an output multiplexer; and

transferring said plurality of bytes in said delay register to said output multiplexer.

27. The method of claim 26 wherein said steps of coupling predetermined bytes of a second plurality of data bytes to an output multiplexer and transferring said first plurality of bytes in said delay register to said output multiplexer are performed simultaneously.

28. The method of claim 26 further comprising a step of providing an interface circuit coupled to said output multiplexer.

29. The method of claim 28 further comprising a step of concatenating data bytes from a first group of data and a second group of data at said user interface circuit.

30. The method of claim 29 further comprising a step of outputting said data from said interface circuit when said interface circuit is full.